**Military College of Signals  
 Mid Term Exam BESE-15 A   
Data Structure & Algorithms**

**Instructor : Bilal Rauf Time = 90 Minutes  
 Max Marks: 50**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Question 1:[Analysis of Algorithm] [Marks=2+2+3]**

1. Explain analysis of Algorithm briefly. What are the factors affecting the running time of an algorithm?
2. Give the worst case time complexity of the following procedure as a function of *n* by using "Big-Oh" notation. Show all the steps of calculation.
   1. **void func\_a(int arr[], int n) // n is size of arr**

**{**

**cout << "Item 0 = " << arr[0] << endl;**

**cout << "Item 1 = " << arr[1] << endl;**

**cout << "Item 2 = " << arr[2] << endl;**

**}**

**[ 2 ]. void func\_b(int arr[], int n) // n is size of arr**

**{**

**for (int a = 0; a < n; ++a)**

**for (int b = 0; b < a; ++b)**

**for (int c = 0; c < b; ++c)**

**arr[c] += arr[b];**

**}**

**Question 2:[Stacks] [Marks=3 + 5]**

1. Write an algorithm to convert Infix Expression to Postfix.
2. Convert the expression to equivalent Postfix notations.Show all the steps involved in conversion
   1. **J – K / ( G ^ H ) + ( N + M )**

**Question 3:[Queues] [Marks = 2 + 7]**

1. Define Priority Queue. Differentiate between Ascending and Descending Priority Queue.
2. Consider a Priority Queue where insertion in the queue is random. Your task is to write a function of **DeQueue\_All\_Highest**, where all nodes with highest priority node should be deleted. You will first find the highest priority in the queue and delete all nodes containing highest priority. A node structure is defined as follows

struct **QNode**{  
 int data;  
 int priority;  
 node \* next;

};

**Question 4:[Linked List] [Marks = 6+ 4]**

1. Suppose we have a singly linked list with random duplicate elements as shown in figure below. Your task is to remove the duplicate elements from the list. To do this task, you have to write a function called **RemoveDuplicate** to remove all the duplicate elements of the list. The list should have unique elements. The structure definition of the linked list is shown below

struct **List**{  
 int Number;  
 List \*Next;  
}\* head;

**Before the Call of Function**

  
**After the Call of Function**



1. Write a function called **"SwapNodes**". This function will swap first and last nodes of the linked list. Use same structure definition as shown in part B.